REMARKS/ARGUMENTS

Reconsideration and allowance of this application are respectfully requested. Currently, claims 1-18 are pending in this application.

Section 3 (page 2) of the Office Action states "Claims 8-9 and 14-15 are rejected under 35 U.S.C. 10*2(b)* as being unpatentable over Rosenberg et al. (United States Patent 5,734,373), hereinafter referenced as 373, in view of Braun et al. (United States Patent 6,411,276), hereinafter referenced as Braun (emphasis added)." Applicant traverses this rejection.

Anticipation under Section 102 of the Patent Act requires that a prior art reference disclose every claim element of the claimed invention. See, e.g., *Orthokinetics, Inc. v. Safety Travel Chairs, Inc.*, 806 F.2d 1565, 1574 (Fed. Cir. 1986). Section 3 (page 3) of the Office Action later openly admits "However 373 fails to disclose transmission via a connectionless network." Given this admission by the Office Action, claims 8-9 and 14-15 are clearly not anticipated under 35 U.S.C. §102. That is, the above-noted rejection of claims 8-9 and 14-15 under 35 U.S.C. §10*2* is deficient on its face. Applicant therefore requests that the rejection of claim 8-9 and 14-15 be withdrawn.

As noted above, section 3 (page 3) of the Office Action admits that "However 373 fails to disclose transmission via a connectionless network." Section 3 (page 3) of the Office Action later alleges that Braun resolves this admitted deficiency of 373. For example, page 3 of the Office Action alleges "In a similar field of invention Braun discloses transmission via a connectionless network (column 4 lines 61-67; column 5 lines 1-2; figure 1)." Applicant respectfully disagrees with this allegation. In particular, col. 4, line 61 to col. 5, line 2 (specifically identified by the Office Action) states the following:

An interface device 14 is coupled to host computer system 12 by a bi-directional bus 24. The bi-directional bus sends signals in either direction between host computer system 12 and the interface device, and can be a serial bus, parallel bus, Universal Serial Bus (USB), Firewire (IEEE 1394) bus, wireless communication interface, etc. An interface port of host computer system 12, such as an RS232 or Universal Serial Bus (USB) serial interface port, parallel port, game port, etc., connects bus 24 to host computer system 12.

Fig. 1 (also specifically identified by the Office Action) is reproduced below:

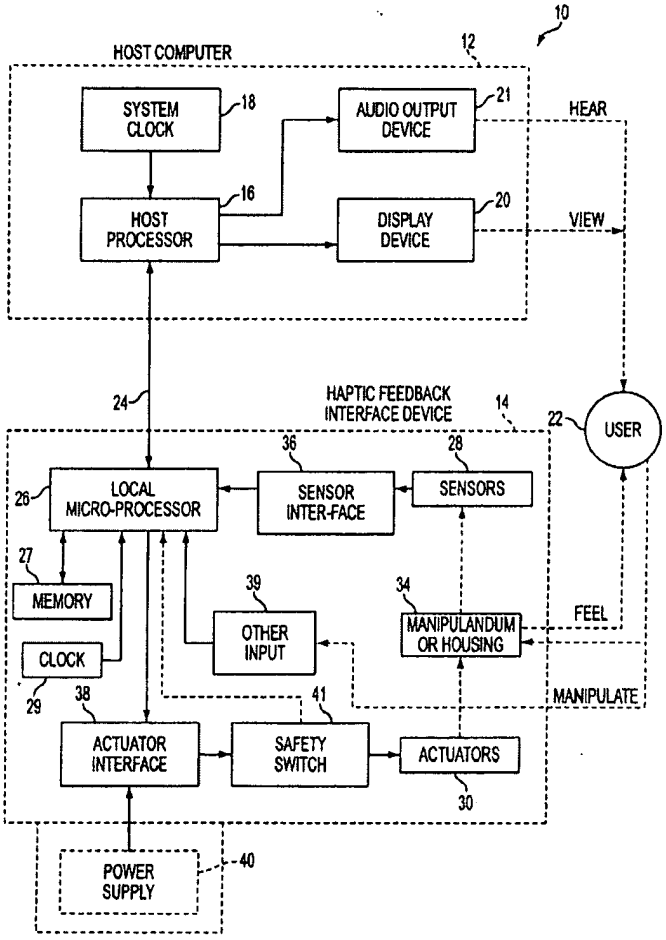


FIG. 1

The above passage (col. 4, line 61 to col. 5, line 2) and Fig. 1 of Braun disclose a host computer 12 and an interface device 14. The host computer 12 and the interface device 14 are connected by a bi-directional bus 24. However, Braun's system comprising the host computer

12 connected to the interface device 14 through the bi-directional bus 24 is not a connectionless network as alleged by the Office Action.

A connectionless link is the antithesis of a connection-oriented, point-to-point link in that there is no pre-determined, dedicated signal path. See, e.g., page 1 lines 20-22 of the originally-filed specification.[1]

A bus connection of the type described by Braun's bi-directional bus 24 is a connection-oriented physical link which carries signals, in this case, between the host computer system 12 and the interface device 14.[2]

Braun's bi-directional bus 24, which specifically connects the host computer system 12 and the interface device 14 in a specific point to point manner, is therefore not connectionless. Accordingly, Braun's system illustrated in Fig. 1 (including the bi-directional bus 24 connecting the host computer system 12 and the interface device 14) is not a connectionless network.

It is entirely logical that the connection between the host computer system 12 and interface device 14 in Braun takes the form of a bus (i.e. a dedicated signal path comprising a pre-determined link (not a network) with specific origin and destination points). This is because Braun discloses what is essentially a haptic interface feedback system 10 (see Figure 1 reproduced above and description in column 4 lines 15-16) which is sited at a single location

---

[1] Also, see the definitions (copy provided as Attachment A to this response) provided by the following sources:

(a) http://en.wikipedia.org/wiki/Connectionless_protocol, and

(b) http://www.yourdictionary.com/telecom/connection-oriented

[2] See the description of a "bus" (copy provided as Attachment B to this response) provided in the following source:

http://en.wikipedia.org/wiki/Bus_(computing)#Description_of_a_bus

proximate to the user 22. Referring to Figure 1, this proximity to the host computer system 12 and interface device 14 allows the user to "hear" and "view" from the output devices 20 and 21 being part of the host computer system 12, and to "feel" and "manipulate" the I/O devices 34 and 39 which are parts of the interface device 14. There is therefore no reason to interpose a connectionless network between the devices making up the system. Accordingly, Braun's disclosed structure (bus 24 specifically connecting system 12 and device 14 as source or destination points) and its associated benefits (see discussion of proximity above) actually teach away from a connectionless network.

Accordingly, Applicant respectfully requests that the rejection of claims 8-9 and 14-15 under 35 U.S.C. §102 be withdrawn.

Claims 1-3, 6 and 13 were rejected under 35 U.S.C. §103 as allegedly being unpatentable over Rosenberg et al. (U.S. '530, hereinafter "Rosenberg '530") in view of Rosenberg '373, and further in view of Braun. Applicant traverses this rejection.

In order to establish a *prima facie* case of obviousness, all of the claim limitations must be taught or suggested by the prior art. The three-way combination of Rosenberg '530, Rosenberg '373 and further in view of Braun fails to teach or suggest all of the claim limitations. For example, the combination fails to teach or suggest "a method of activating a haptic output device of the kind responsive to signals defining directional force comprising receiving a series of signals defining a multiplicity of data packets, each packet defining a directional force applied at one location for transmission to the current location via a *connectionless network*...(emphasis added)" as required by independent claim 1 and its dependents.

Section 4 (page 6) admits that "However 530 fails to disclose transmission via a connectionless network...." For the reasons discussed above, Braun fails to resolve this admitted

deficiency of Rosenberg '530. Again, contrary to the allegations of the Office Action, Braun fails to disclose or even suggest a connectionless network. Accordingly, even if the teachings of Rosenberg '530, Rosenberg '373 and Braun were combined as proposed by the Office Action, the combination would not have been taught or suggested all of the claim limitations.

Claim 4 was rejected under 35 U.S.C. §103 as allegedly being unpatentable over the four-way combination of Rosenberg '530, in view of Rosenberg '373, and Braun, and further in view of Chafe. Claim 5 was rejected under 35 U.S.C. §103 as allegedly being unpatentable over the five-way combination of Rosenberg '530, in view of Rosenberg '373 and Braun, and in view of Chafe, and further in view of Wang et al. Claim 7 was rejected under 35 U.S.C. §103 as allegedly being unpatentable over the four-way combination of Rosenberg '530, in view of Rosenberg '373 and Braun, and further in view of Wang. Claims 10-11 and 16-17 were rejected under 35 U.S.C. §103 as allegedly being unpatentable over the four-way combination of Rosenberg'373 in view of Braun and in view of Niemeyer et al. Claims 12 and 18 were rejected under 35 U.S.C. §103 as allegedly being unpatentable over the four-way combination of Rosenberg '373 in view of Braun and in view of Wang and further in view of Rosenberg '613. Applicant traverses each of these rejections. In particular, Applicant submits that none of these third, fourth and/or fifth cited references resolve the above-described deficiencies with respect to base independent claims 1, 8 or 14, each of which requires a connectionless network. Accordingly, Applicant respectfully requests that each of the rejections under 35 U.S.C. §103 be withdrawn.

Conclusion:

Applicant believes that this entire application is in condition for allowance and respectfully requests a notice to this effect. If the Examiner has any questions or believes that an

interview would further prosecution of this application, the Examiner is invited to telephone the undersigned.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: _____
Raymond Y. Mah
Reg. No. 41,426

RYM:dmw
901 North Glebe Road, 11th Floor
Arlington, VA 22203-1808
Telephone: (703) 816-4000
Facsimile: (703) 816-4100

APPENDIX A:

http://en.wikipedia.org/wiki/Connectionless_protocol

http://www.yourdictionary.com/telecom/connection-oriented

Connectionless protocol

From Wikipedia, the free encyclopedia

In telecommunications, **connectionless** describes communication between two network end points in which a message can be sent from one end point to another without prior arrangement. The device at one end of the communication transmits data to the other, without first ensuring that the recipient is available and ready to receive the data. The device sending a message simply sends it addressed to the intended recipient. As such there are more frequent problems with transmission than with connection-oriented protocols and it may be necessary to resend the data several times. Connectionless protocols are often disfavored by network administrators because it is much harder to filter malicious packets from a connectionless protocol using a firewall. The Internet Protocol (IP) and User Datagram Protocol (UDP) are connectionless protocols, but TCP/IP (the most common use of IP) is connection-oriented.

Connectionless protocols are usually described as stateless because the endpoints have no protocol-defined way to remember where they are in a "conversation" of message exchanges. The alternative to the connectionless approach uses connection-oriented protocols, which are sometimes described as stateful because they can keep track of a conversation.

List of connectionless protocols

- IP (internet layer, can also be used for connections)
- UDP
- ICMP
- IPX
- TIPC
- NetBEUI

See also

- Connection-oriented protocol

connection-oriented

connection-oriented definition - telecom

A transmission mode in which a network establishes a logical relationship, and often a predetermined path, for all frames or packets associated with an originating and destination address pair to travel.The path can be permanent or can exist for a single session.The public switched telephone network (PSTN) is connection-oriented.Transmission Control Protocol (TCP) is a connection-oriented protocol, where Internet Protocol (IP) is a connectionless datagram service. See also connectionless, frame, IP, packet, protocol, PSTN, session, and TCP.

comments

Improve this definition.
Do you have more to add? Share your linguistic knowledge or observation.
Name: anonymous Login/Register to save your comments.

APPENDIX B:

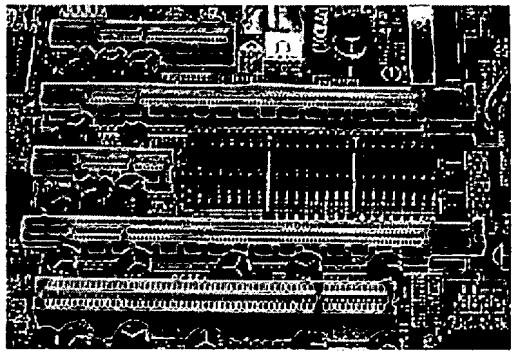
http://en.wikipedia.org/wiki/Bus_(computing)#Description_of_a_bus

Bus (computing)

From Wikipedia, the free encyclopedia

In computer architecture, a **bus** is a subsystem that transfers data between computer components inside a computer or between computers.

Early computer buses were literally parallel electrical buses with multiple connections, but the term is now used for any physical arrangement that provides the same logical functionality as a parallel electrical bus. Modern computer buses can use both parallel and bit-serial connections, and can be wired in either a multidrop (electrical parallel) or daisy chain topology, or connected by switched hubs, as in the case of USB.



4 PCI Express bus card slots (from top to bottom: x4, x16, x1 and x16), compared to a traditional 32-bit PCI bus card slot (very bottom).

Contents

History

First generation

Early computer buses were bundles of wire that attached memory and peripherals. They were named after electrical buses, or busbars. Almost always, there was one bus for memory, and another for peripherals, and these were accessed by separate instructions, with completely different timings and protocols.

One of the first complications was the use of interrupts. Early computer programs performed I/O by waiting in a loop for the peripheral to become ready. This was a waste of time for programs that had other tasks to do. Also, if the program attempted to perform those other tasks, it might take too long for the program to check again, resulting in loss of data. Engineers thus arranged for the peripherals to interrupt the CPU. The interrupts had to be prioritized, because the CPU can only execute code for one peripheral at a time, and some devices are more time-critical than others.

Later computer programs began to share memory common to several CPUs. Access to this memory bus had to be prioritized, as well.

The classic, simple way to prioritize interrupts or bus access was with a daisy chain.

DEC noted that having two buses seemed wasteful and expensive for mass-produced minicomputers, and mapped peripherals into the memory bus, so that the devices appeared to be memory locations.

Early microcomputer bus systems were essentially a passive backplane connected directly or through buffer amplifiers to the pins of the CPU. Memory and other devices would be added to the bus using the same address and data pins as the CPU itself used, connected in parallel. Communication was controlled by the CPU, which had read and written data from the devices as if they are blocks of memory, using the same instructions, all timed by a central clock controlling the speed of the CPU. Still, devices interrupted the CPU by signaling on separate CPU pins. For instance, a disk drive controller would signal the CPU that new data was ready to be read, at which point the CPU would move the data by reading the "memory location" that corresponded to the disk drive. Almost all early microcomputers were built in this fashion, starting with the S-100 bus in the Altair.

In some instances, most notably in the IBM PC, although similar physical architecture is employed, instructions to access peripherals (`in` and `out`) and memory (`mov` and others) have not been made uniform at all, and still generate distinct CPU signals, that could be used to implement a separate I/O bus.

These simple bus systems had a serious drawback when used for general-purpose computers. All the equipment on the bus has to talk at the same speed, as it shares a single clock.

Increasing the speed of the CPU becomes harder, because the speed of all the devices must increase as well. When it is not practical or economical to have all devices as fast as the CPU, the CPU must either enter a wait state, or work at a slower clock frequency temporarily[1], to talk to other devices in the computer. While acceptable in embedded systems, this problem was not tolerated for long in general-purpose, user-expandable computers.

Such bus systems are also difficult to configure when constructed from common off-the-shelf equipment. Typically each added expansion card requires many jumpers in order to set memory addresses, I/O addresses, interrupt priorities, and interrupt numbers.

A *bus controller* accepted data from the CPU side to be moved to the peripherals side, thus shifting the communications protocol burden from the CPU itself. This allowed the CPU and memory side to evolve separately from the device bus, or just "bus". Devices on the bus could talk to each other with no CPU intervention. This led to much better "real world" performance, but also required the cards to be much more complex. These buses also often addressed speed issues by being "bigger" in terms of the size of the data path, moving from 8-bit parallel buses in the first generation, to 16 or 32-bit in the second, as well as adding software setup (now standardised as Plug-n-play) to supplant or replace the jumpers.

However these newer systems shared one quality with their earlier cousins, in that everyone on the bus had to talk at the same speed. While the CPU was now isolated and could increase speed without fear, CPUs and memory continued to increase in speed much faster than the buses they talked to. The result was that the bus speeds were now very much slower than what a modern system needed, and the machines were left starved for data. A particularly common example of this problem was that video cards quickly outran even the newer bus systems like PCI, and computers began to include AGP just to drive the video card. By 2004 AGP was outgrown again by high-end video cards and is being replaced with the new PCI Express bus.

An increasing number of external devices started employing their own bus systems as well. When disk drives were first introduced, they would be added to the machine with a card plugged into the bus, which is why computers have so many slots on the bus. But through the 1980s and 1990s, new systems like SCSI and IDE were introduced to serve this need, leaving most slots in modern systems empty. Today there are likely to be about five different buses in the typical machine, supporting various devices.

Third generation

"Third generation" buses have been emerging into the market since about 2001, including HyperTransport and InfiniBand. They also tend to be very flexible in terms of their physical connections, allowing them to be used both as internal buses, as well as connecting different machines together. This can lead to complex problems when trying to service different requests, so much of the work on these systems concerns software design, as opposed to the hardware itself. In general, these third generation buses tend to look more like a network than the original concept of a bus, with a higher protocol overhead needed than early systems, while also allowing multiple devices to use the bus at once.

Buses such as Wishbone have been developed by the open source hardware movement in an attempt to further remove legal and patent constraints from computer design.

Description of a bus

At one time, "bus" meant an electrically parallel system, with electrical conductors similar or identical to the pins on the CPU. This is no longer the case, and modern systems are blurring the lines between buses and networks.

Buses can be parallel buses, which carry data words in parallel on multiple wires, or serial buses, which carry data in bit-serial form. The addition of extra power and control connections, differential drivers, and data connections in each direction usually means that most serial buses have more conductors than the minimum of one used in the 1-Wire and UNI/O serial buses. As data rates increase, the problems of timing skew, power consumption, electromagnetic interference and crosstalk across parallel buses become more and more difficult to circumvent. One partial solution to this problem has been to double pump the bus. Often, a serial bus can actually be operated at higher overall data rates than a parallel bus, despite having fewer electrical connections, because a serial bus inherently has no timing skew or crosstalk. USB, FireWire, and Serial ATA are examples of this. Multidrop connections do not work well for fast serial buses, so most modern serial buses use daisy-chain or hub designs.

Most computers have both internal and external buses. An *internal bus* connects all the internal components of a computer to the motherboard (and thus, the CPU and internal memory). These types of buses are also referred to as a local bus, because they are intended to connect to local devices, not to those in other machines or external to the computer. An *external bus* connects external peripherals to the motherboard.

Network connections such as Ethernet are not generally regarded as buses, although the difference is largely conceptual rather than practical. The arrival of technologies such as InfiniBand and HyperTransport is further blurring the boundaries between networks and buses. Even the lines between internal and external are sometimes fuzzy, I^2C can be used as both an internal bus, or an external bus (where it is known as ACCESS.bus), and InfiniBand is intended to replace both internal buses like PCI as well as external ones like Fibre Channel. In the typical desktop application, USB serves as a peripheral bus, but it also sees some use as a networking utility and for connectivity between different computers, again blurring the conceptual distinction.

Bus topology

In a network, the master scheduler controls the data traffic. If data is to be transferred, the requesting computer sends a message to the scheduler, which puts the request into a queue. The message contains an identification code which is broadcast to all nodes of the network. The scheduler works out priorities and notifies the receiver as soon as the bus is available.

The identified node takes the message and performs the data transfer between the two computers. Having completed the data transfer the bus becomes free for the next request in the scheduler's queue.

- Advantage: Any computer can be accessed directly and messages can be sent in a relatively simple and fast way.
- Disadvantage: A scheduler is required to organize the traffic by assigning frequencies and priorities to each signal.

See also: Bus network.

Examples of internal computer buses

Parallel

- ASUS Media Bus proprietary, used on some ASUS Socket 7 motherboards
- Computer Automated Measurement and Control (CAMAC) for instrumentation systems
- Extended ISA or EISA
- Industry Standard Architecture or ISA
- Low Pin Count or LPC
- MBus
- MicroChannel or MCA
- Multibus for industrial systems
- NuBus or IEEE 1196
- OPTi local bus used on early Intel 80486 motherboards.
- Conventional PCI
- Q-Bus, a proprietary bus developed by Digital Equipment Corporation for their PDP and later VAX computers.
- S-100 bus or IEEE 696, used in the Altair and similar microcomputers
- SBus or IEEE 1496
- STD Bus (for STD-80 [8-bit] and STD32 [16-/32-bit]), FAQ (http://www.controlled.com/std/faq.html)
- Unibus, a proprietary bus developed by Digital Equipment Corporation for their PDP-11 and early VAX computers.
- VESA Local Bus or VLB or VL-bus
- VMEbus, the VERSAmodule Eurocard bus

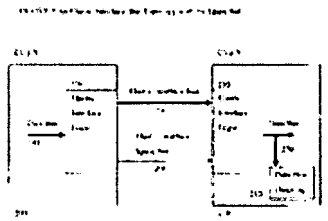
Serial

- 1-Wire
- HyperTransport
- I^2C
- PCI Express or PCIe
- Serial Peripheral Interface Bus or SPI bus

- UNI/O

Self Repairable

Self repairable elastic interface buses have recently been invented by IBM. IBM has filed a patent application on these buses which is undergoing peer review on Peer to Patent. The public commentary period closed on July 24, 2008.[2] The IBM invention provides a spare net which the system switches to in the event that an alternate net doesn't function.



Spare Net for Elastic Interface Bus from US patent application 20080082878[2]

Examples of external computer buses

Parallel

- Advanced Technology Attachment or ATA (aka PATA, IDE, EIDE, ATAPI, etc.) disk/tape peripheral attachment bus
  (the original ATA is parallel, but see also the recent serial ATA)
- HIPPI HIgh Performance Parallel Interface
- IEEE-488 (aka GPIB, General-Purpose Interface Bus, and HPIB, Hewlett-Packard Instrumentation Bus)
- PC card, previously known as *PCMCIA*, much used in laptop computers and other portables, but fading with the introduction of USB and built-in network and modem connections
- SCSI Small Computer System Interface, disk/tape peripheral attachment bus

Serial

- USB Universal Serial Bus, used for a variety of external devices
- Serial Attached SCSI and other serial SCSI buses
- serial ATA
- Controller Area Network ("CAN bus")
- EIA-485
- FireWire

Examples of internal/external computer buses

- Futurebus

- InfiniBand
- QuickRing
- SCI

See also

- Address bus
- Bus contention
- Control bus
- Front side bus
- Network On Chip
- List of device bandwidths

References

1. ^ Bray, Andrew C.; Dickens, Adrian C.; Holmes, Mark A. (1983). "28. The One Megahertz bus (http://www.nvg.org/bbc/doc/BBCAdvancedUserGuide-PDF.zip) " (zipped PDF). *The Advanced User Guide for the BBC Microcomputer*. Cambridge, UK: Cambridge Microcomputer Centre. pp. 442–443. ISBN 0946827001. http://www.nvg.org/bbc/doc/BBCAdvancedUserGuide-PDF.zip. Retrieved 2008-03-28.
2. ^ *a b* Peer to Patent review page for "System and Method to Support Use of Bus Spare Wires in Connection Modules" (http://www.peertopatent.org/patent/20080082878/overview)

External links

- Chip Weems' Lecture 12: Buses (http://www.cs.umass.edu/~weems/CmpSci635/635lecture12.html)
- Computer hardware buses (http://www.dmoz.org/Computers/Hardware/Buses//) at the Open Directory Project
- Computer hardware buses and slots pinouts with brief descriptions (http://pinouts.ru/pin_Slots.shtml)

Retrieved from "http://en.wikipedia.org/wiki/Bus_(computing)"
Categories: Digital electronics | Computer buses | Motherboard